First Order Logic

ITI0210, lecture 8 (2021)

Review

Entailment: $KB \models \alpha$

using existing **knowledge** to deduce e.g. answers to questions

Truth value algorithms:

- truth table model checking
- DPLL

Syntax based inference:

- Forward chaining on Horn clauses
- Resolution

Motivation

In first order logic (FOL):

$$Pit(X) \wedge NextTo(Y, X) \rightarrow Breeze(Y)$$

In propositional logic:

$$Pit_{1,1} \rightarrow Breeze_{2,1}$$
$$Pit_{1,1} \rightarrow Breeze_{1,2}$$
$$Pit_{2,1} \rightarrow Breeze_{1,1}$$
$$Pit_{2,1} \rightarrow Breeze_{2,2}$$
$$Pit_{2,1} \rightarrow Breeze_{3,1}$$

etc…

Motivation

In first order logic (FOL):

$$Brother(Y, X) \rightarrow Male(Y)$$

In propositional logic, not really possible

(unless we pick a subset of people and generate rules for them)

This type of **generalization** is key in A.I.

FOL Syntax and Semantics

An informal introduction

Some Examples

$Bird(tweety)$ - Tweety is a bird

$Bird(X) \rightarrow Flies(X)$ – If X is a bird, it flies

$\forall X\ Bird(X) \rightarrow Flies(X)$ – same thing more formally

$\exists X\ AtStanford(X) \wedge Student(X) \wedge Smart(X)$ – there is a student at Stanford who is smart

Some Examples

Predicate

Constant

Variable

Quantifier

$Bird(tweety)$ - Tweety is a bird

$Bird(X) \rightarrow Flies(X)$ – If X is a bird, it flies

$\forall X\ Bird(X) \rightarrow Flies(X)$ – same thing more formally

$\exists X\ AtStanford(X) \wedge Student(X) \wedge Smart(X)$ – there is a student at Stanford who is smart

More Examples

Multiple variables

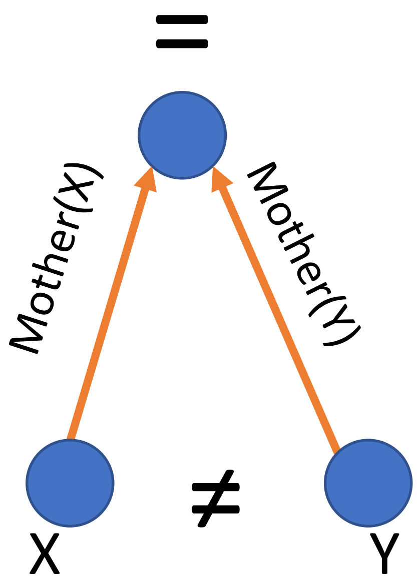
$$Smokes(X) \wedge Influences(X,Y) \rightarrow Smokes(Y)$$

Functions

$$Human(X) \rightarrow Human\big(Mother(X)\big)$$

Equality

$$Girl(X) \wedge Girl(Y) \wedge Mother(X) = Mother(Y) \wedge \neg(X = Y) \rightarrow Sister(X,Y)$$

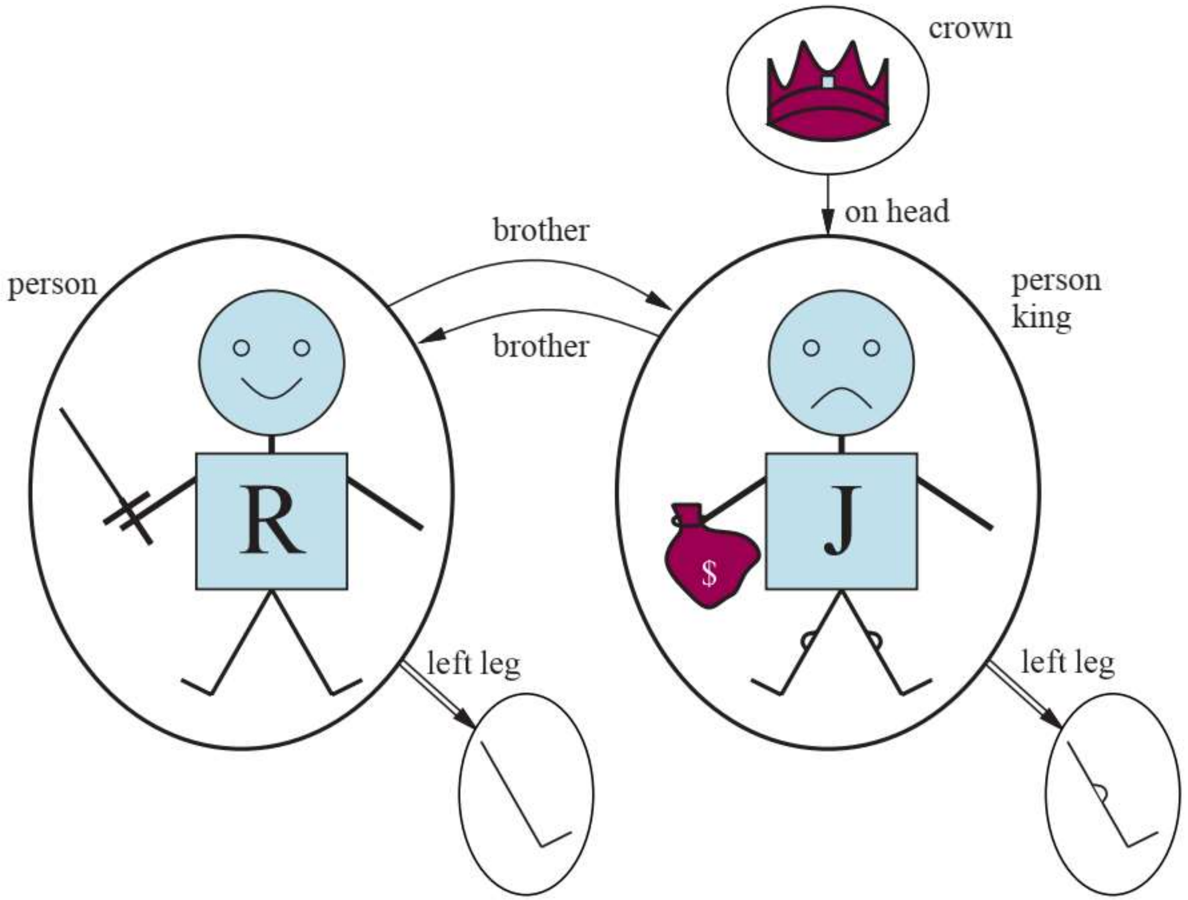


Semantics

Semantics, or meaning of sentences/formulas

Everything refers to an object, relation, function

Each sentence has a truth value (like in propositional logic)



Semantics

facts:

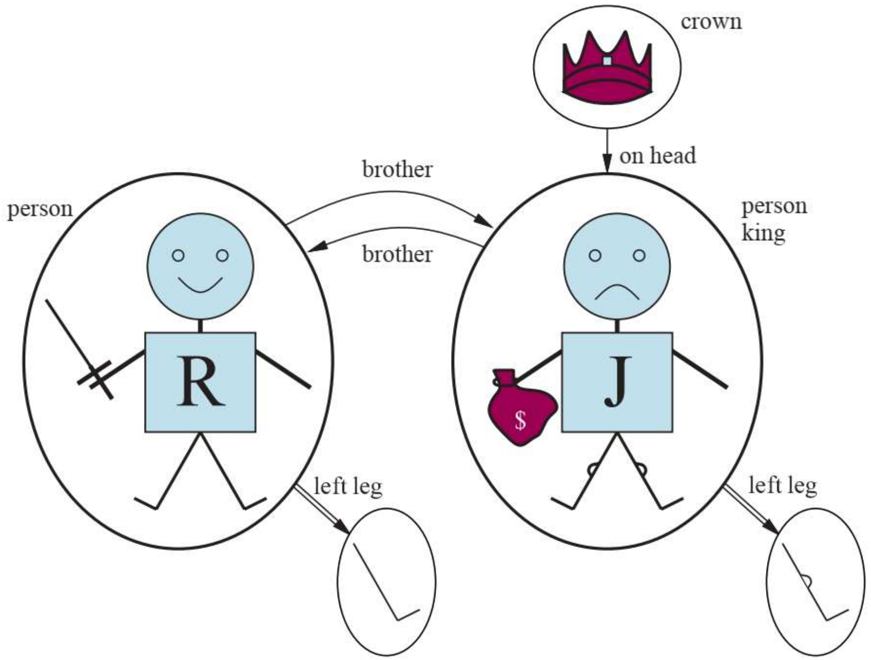
$$Person(richard)$$
$$Person(john)$$
$$Brother(richard, john)$$
$$Brother(john, richard)$$
$$King(john)$$

reasoning about royal headgear:

$$King(X) \land Onhead(X, Y) \rightarrow Crown(Y)$$

each brother has their own left leg:

$$Brother(X, Y) \rightarrow \neg(LeftLeg(X) = LeftLeg(Y))$$



Semantics: each predicate is about some relation, each constant/variable/function about an object

Semantics

Each formula has a **truth value**, because FOL formulas consist of:

Terms: $Function(term_1, \ldots, term_n)$ or $constant$ or $variable$

Atomic sentence: $Predicate(term_1, \ldots, term_n)$

Predicates are always True or False

Construct recursively from atomic sentences:

if S_1 and S_2 are sentences, then

$\neg S_1,\ S_1 \vee S_2,\ S_1 \wedge S_2,\ S_1 \rightarrow S_2,\ S_1 \Leftrightarrow S_2$ are sentences

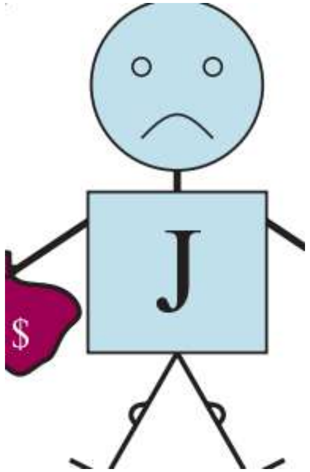
Can compute truth values as each S is True/False

Semantics

Fun fact about semantics:

The reasoner has no idea about the intended semantics

The constant “richard” could mean this guy too



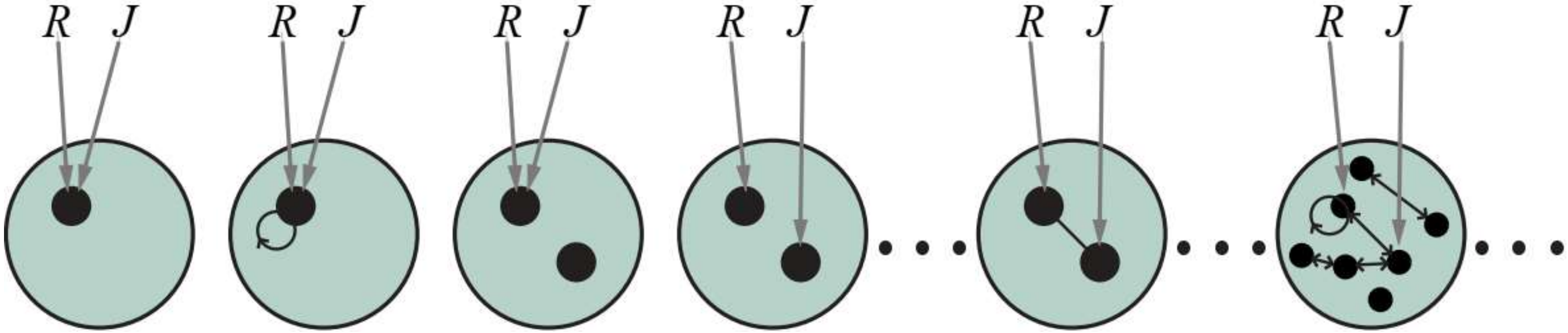
This is why model checking is not possible in FOL
(Can’t test every possible value of symbols)

FOL Inference

Deducing things from FOL knowledge

Deciding Entailment

Generally, not possible to decide $KB \vDash \alpha$
(too many/infinite number of models to check)



Pictured: models for two constant symbols and one relation

Usable Inference Methods

High end theorem provers:

- full first order logic
- resolution

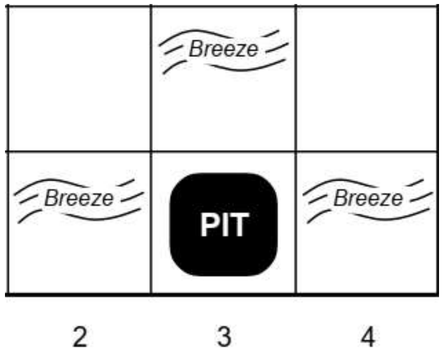
PROLOG:

- simplified language
- depth first search

Resolution in FOL

Handle variables and constants by **unification**:

$$\frac{\neg Pit(X) \lor Dangerous(X) \quad Pit(square_{3,1})}{Dangerous(square_{3,1})}$$



(substitution $\{X/square_{3,1}\}$ applied to input clauses)

Accessible explanation in Estonian (not part of the course):

Tamme, T., Tammet, T., Prank, R. Loogika: mõtlemisest tõestamiseni, TÜ Kirjastus, 1997
https://dspace.ut.ee/handle/10062/24397

Knowledge Representation

The main use for FOL in AI

Facts and Rules

“Grounded” facts, about specific entities

$$Smokes(bob)$$
$$Influencer(bob)$$
$$Roommate(bob, carl)$$

“Lifted” rules, allow generalized reasoning

$$Smokes(X) \wedge Influencer(X) \wedge Roommate(X, Y) \rightarrow Smokes(Y)$$

Quantifiers

You probably noted chronic lack of $\forall, \exists$ quantifiers on the slides

$$Bird(X) \rightarrow Flies(X)$$

$\forall X$ is implied here (the common sense way to read the rule)

The existential quantifier cannot be omitted

$$\exists X\ Loves(X, alice)$$

But, an anonymous constant will encode the same knowledge

$$Loves(anon1, alice)$$

Implementations

FOL:
https://en.wikipedia.org/wiki/Otter_(theorem_prover)
https://github.com/tammet/gkc
https://vprover.github.io/

PROLOG:
http://www.gprolog.org/
https://www.swi-prolog.org/

Each has **different syntax, capabilities** etc.
Lots of books about PROLOG!

Open World

Test this input using http://logictools.org/

```
-innocent(X) => guilty(X).
innocent(bob).
% negated query
guilty(bob).
```

What happens? What did you expect?

Open World

What about this:

```
earthquake(X) => alarm(X).
burglary(X) => alarm(X).
-burglary(today).
-earthquake(today).
alarm(today).
% or try this
% -alarm(today).
```

Not able to prove “Some statement”

does **not** mean “Some statement is false”

Open World

Gandalf revisited

Wizards wear a wizard hat
Gandalf wears a wizard hat

```
WizardHat(gandalf).
Wizard(X) => WizardHat(X).
Wizard(X) => $ans(X).
```

Turns out Gandalf wasn't a wizard after all?